International Journal on Artificial Intelligence Tools
Vol. 22, No. 5 (2013) 1360004 (21 pages)
© World Scientific Publishing Company
DOI: 10.1142/S021821301360004X

World Scientific
www.worldscientific.com

AUTOMATIC PEDESTRIAN SEGMENTATION COMBINING SHAPE, PUZZLE AND APPEARANCE

YANLI LI, ZHONG ZHOU* and WEI WU

State Key Laboratory of Virtual Reality Technology and Systems,
Beihang University, Beijing, 100191, P. R. China
zz@vrlab.buaa.edu.cn

Received 13 November 2012
Accepted 4 March 2013
Published 25 October 2013

In this paper, we address the problem of automatically segmenting non-rigid pedestrians in still images. Since this task is well known difficult for any type of model or cue alone, a novel approach utilizing shape, puzzle and appearance cues is presented. The major contribution of this approach lies in the combination of multiple cues to refine pedestrian segmentation successively, which has two characterizations: (1) a shape guided puzzle integration scheme, which extracts pedestrians via assembling puzzles with constraint of a shape template; (2) a pedestrian refinement scheme, which is fulfilled by optimizing an automatically generated trimap that encodes both human silhouette and skeleton. Qualitative and quantitative evaluations on several public datasets verify the approach's effectiveness to various articulated bodies, human appearance and partial occlusion, and that this approach is able to segment pedestrians more accurately than methods based only on appearance or shape cue.

Keywords: Pedestrian segmentation; KDE-EM; shape matching; puzzle integration; appearance trimap.

1. Introduction

Pedestrians, as the principal actors in daily life, have been widely studied in computer vision. The goal of pedestrian segmentation is to provide a precise and complete body mask, which is a fundamental task for many artificial intelligence applications like action recognition, human-computer interaction, image retrieval, photo summarization and so on. At the same time, this has proven to be a challenging task since that: (1) the human shapes undergo a large variety of transformations due to body articulations; (2) the backgrounds tend to be cluttered and have similar color or texture with the foreground bodies; (3) various types and styles of clothes

*Corresponding author

result in a large variety of body appearance; (4) human bodies may be occluded by accessories, street objects or other persons.

Among the developed approaches, shape, puzzle and appearance are mainly used cues. Shape is characterized as one-dimensional curve, thus is invariant to lighting conditions and object colors. But a conventional shape matching method[1] is sensitive to cluttered backgrounds. Boundary points on a shape template are often misaligned with edge apart from body contour. Puzzle, also known as superpixel, is formulated by aggregating pixels into a group. It is a more productive unit than pixel and can preserve object boundaries. Based on it, many object segmentation methods[2,3] perform figure extraction within a conditional random field (CRF). Although those methods can regularize and smooth segmentation, they face difficulties with faint body parts like limbs since superpixels may merge foreground regions with the backgrounds. Appearance has the advantage of preserving relative uniform color or texture information for a single object. The interactive segmentation schemes, e.g., GrabCut,[4] which build mainly on appearance cue, have become very popular due to their efficiency in handling various cases. However, without constraint of high-level priors, low-level segmentation methods[5,6] tend to over- or under-segment pedestrians.

The difficulty of pedestrian segmentation makes it problematic to rely on any type of model or feature alone. Following this principle, we draw from the strengths of multiple cues and develop an approach combining high-level shape, mid-level puzzle and low-level appearance cues. The approach works by several stages. The input to the approach are pedestrian windows produced by a pedestrian detector.[7] For each candidate window, a KDE-EM (Kernel Density Estimation-Expectation Maximization) scheme is first employed to estimate the probabilities of pixels belonging to the foreground. The limitation with this stage is that it fails to produce clear contours. Therefore, at stage II, we divide the detection window into small puzzles, and reassemble the puzzles into a human figure under the guide of a shape template. Since hundreds of shape templates need to be visited at this stage, we organize them into a hierarchical tree for speeding up. The segmentation problem is formulated as a Markov Random Field(MRF) energy minimization. We design a greedy solver to optimize it. We further refine body boundaries via optimizing an appearance trimap. The trimap which indicates the inside, outside and unknown regions of human body is generated by human silhouette and skeleton. Pedestrian refinement is performed by optimizing the unknown regions with a global energy function. Figure 1 gives a high-level overview of our approach.

The major contribution of this approach lies in the combination of multiple cues to refine pedestrian segmentation successively. In contrast to previous methods, this approach has the following characterizations: (1) a shape guided puzzle integration scheme which extracts pedestrians via assembling puzzles with constraint of a shape template, and thus preserves human boundaries; (2) a pedestrian refinement scheme which is fulfilled with an automatically generated trimap. The trimap encodes both human silhouette and skeleton, and hence can produce human-like segmentation.

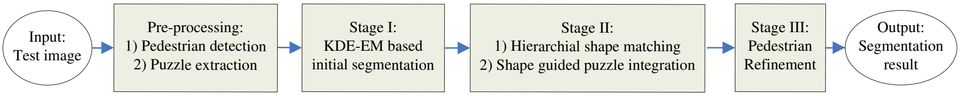


Fig. 1. The framework of the approach. The input is an image with several pedestrians. After the pedestrians are roughly detected, the foreground probabilities are computed via the KDE-EM scheme, and then pedestrian silhouettes are extracted via shape guided puzzle integration, finally the segmentation is refined through the generated trimap.

According to the processing stages mentioned above, this paper is organized as follows. In Section 2 we summarize the related works. Section 3 presents the KDE-EM based initialization. The details of shape guided puzzle integration are described in Section 4. The pedestrian refinement using an automatically generated trimap is presented in Section 5. Section 6 demonstrates the experimental results. Some conclusions and discussions are given in the last section.

2. Related Works

Over the past decades, numerous approaches have been proposed for pedestrian segmentation. According to the type of cues relied on, these approaches can be roughly grouped into four categories: shape-based, appearance-based, puzzle-based and combined ones.

The first category uses shape as the main discriminative cue, including global shape template and local contour. Methods based on shape template extract pedestrians by matching shape templates with the image's edge map. Effective shape registration plays a central role. For example, Gavrila[1] performs shape matching using Distance Transformation(DT) and Chamfer matching. Active contour models (also called snake, e.g., Ref. 8) try to attach template points to human boundaries by iteratively solving a global energy in level-set space. Although global shape matching methods can handle color variances within the image, they are very sensitive to cluttered backgrounds and occlusion due to shape variances. In contrast, local contours are more flexible, such as the boundary fragments employed in Opelt *et al.*,[9] the pose contours in Lin *et al.*,[10] the edgelets in Wu *et al.*,[11] the adaptive contours in Gao *et al.*[12] Methods based on local contour delineate human boundaries by selection of contour features in a supervised manner. The responses of contours are integrated to locate object centroid, and the back-projection in turn helps to find foreground boundaries. However, the pixel-level segmentation results are unsatisfactory. As shown in Ref. 9, many similar contours are selected around object boundaries, causing ambiguous delineation. In Refs. 10–12, some local contours are misaligned with human boundaries due to the similarity between the background and foreground.

The second category uses appearance to separate human from the backgrounds, involving supervised figure extraction methods[13–16] and unsupervised figure-ground separation methods.[4,17,18] Based on "bag of words", Leibe *et al.*[13] and Wang *et al.*[14]

explore the idea of learning a codebook of appearance parts for interleaved segmentation and detection. They arrange the fragments in star-style, detect human by voting in Hough space, and back-project foreground fragments to formulate pedestrian regions. Another technique is to arrange the learned fragments in CRF and segment foreground objects by solving a global energy function, such as Larlus *et al.*[15] and Gonfaus *et al.*[16] Without constraint of local edge information, these methods fail to extract clear boundaries. More recently, interactive figure-ground separation methods[4,17] and saliency cut methods[18,19] draw lots of attention. Both of them take advantage of local color models to separate foreground objects from the backgrounds. The difference is that the former uses scribbles while the latter uses salience priors to initialize color models. Under constraint of color models, a global energy function in MRF is built for optimizing the foreground objects. However, the interactive methods are cumbersome, while the saliency cut methods are sensitive to cluttered backgrounds.

The third category takes puzzles (superpixels) as the computation units, which encode both contour and appearance information. Pedestrian segmentation based on puzzle study how to combine puzzles to form human body. Taking the similar idea to the supervised figure extraction methods,[15,16] one technique is to arrange puzzles in CRF.[2,20] These approaches first learn the foreground probability of each puzzle, and then group puzzles according to neighboring uniformity. The main difficulty in this technique stems from segmentation granularity. Puzzles tend to merge some background and foreground regions together. Another technique is to select puzzles in a hierarchical segmentation tree to identify body parts.[21,22] In Ref. 21, pools of candidate puzzles are generated from bottom-up segmentation and each puzzle is scored based on its properties. These puzzles are assembled in a bottom-up parse tree which enforces constraints among parts at a given level of the tree and between parents and children. In Ref. 22, candidate puzzles are first classified with a set of local cues, and then global constraints are enforced to sort and complete part configuration. However, such mid-level puzzles may not correspond to semantic body parsing. Consequently, the body parts are often under- or over-segmented.

Drawing advantages of the above categories, some researchers suggest combining shape, puzzle and appearance cues for specific object segmentation. One technique is implemented by grouping the detected appearance fragments under shape guide.[23,24] However, these methods are only suitable for segmenting rigid objects or non-rigid objects with limited shape variation. For highly articulated pedestrians, the limited templates employed in these approaches cannot capture all poses. Comparably, pose-specific MRF methods are more suitable for non-rigid object segmentation, such as ObjCut,[25] PoseCut[26] and Lin *et al.*[27] They are tolerant to pose variances as they consider non-rigid objects as layered pictorial structure with each layer encoding the similar appearance cue. Yet such work faces difficulty with pose parsing, which is still an open issue. Moreover, the iterative optimization in ObjCut[25] makes it computationally expensive, the single stickman employed in PoseCut[26] may not capture various human poses.

Our segmentation approach falls into the last category. Here, motivated by global shape matching method[1] and interactive figure-ground separation approaches,[4,17] we present an automatic pedestrian segmentation approach. Compared to the traditional shape matching approach,[1] our approach performs segmentation by integrating local puzzles, thus is more tolerant to local appearance variances. Comparing with interactive figure-ground separation,[4,17] our approach can automatically extract pedestrian silhouettes and skeletons for segmentation, thus avoids cumbersome manipulation. In contrast to previous combined approaches,[23,24] this approach is characterized by utilization of automatically generated trimaps for human segmentation, which encode the constraint of shape as well as skeleton. We compare our approach with global shape matching method,[1] appearance based methods[4,18] and local contour based methods[10,11] over several public datasets. The experiments demonstrate that our approach improves segmentation significantly and is robust to large variability of human shape, body appearance and partial occlusion.

3. KDE-EM Based Initial Segmentation

Given an input image with multiple pedestrians, we first run a state-of-the-art detector — PFF[7] to find pedestrians. The PFF detector[7] performs quite well for pedestrian detection, being scored top with the PASCAL VOC[a] and INRIA[28] datasets. We enlarge the candidate windows by 10% to include local context. By doing so, the rough locations and scales of the candidate windows will make segmentation easier.

For each pixel x within the window, we wish to label it as the foreground or background. Since the pixels were generated by two stochastic processes — the foreground and background processes, the problem can be considered as assigning each pixel to the process that generated it. If we know the probability density functions of the processes, the problem is to assign each pixel to the process with maximum likelihood. It is well known that there are two ways to define probability density function — the parametric and non-parametric methods. The former uses a particular form for the underlying density, e.g., Gaussian Mixture Model, which is sensitive to initialization and requires selection of the number of mixture components. Comparatively, the latter imposes no formal structure on the data, which is more flexible. In this work, a non-parametric estimator — kernel density estimation (KDE)[29] is adopted.

Let the sample pixel set be $X = \{x_i\}$, each represented by a d-dimensional feature vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, the probability of a new pixel $y = (y_1, y_2, \ldots, y_d)$ from the same distribution of X can be estimated with KDE as:

$$p(y) = \frac{1}{|X|} \sum_{x_i \in X} \prod_{j=1}^{d} ker(y_j - x_{ij}) \tag{1}$$

where $ker(\cdot)$ is the kernel function.

[a] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/

Given the probability density functions of the foreground and background processes, we could assign each pixel to the foreground versus background with maximum likelihood. On the other hand, if we know the assignment of each pixel, we could estimate the probability density functions. Obviously, this is a chicken-and-egg problem. The EM scheme provides a natural way to deal with it. Defining $F^t(x)$ and $B^t(x)$ be the probabilities of the pixel x belonging to the foreground and background at iteration t respectively, $t = 0, \dots, N$, we update segmentation through KDE-EM as follows:

(i) Initialization: The initial probability maps are determined by an offline learned pedestrian shape prior PM (see Fig. 2(b)). That is, $F^0(x) = PM(x)$, $B^0(x) = 1 - PM(x)$. The shape prior $PM(\cdot)$ is obtained via averaging hundreds of pedestrian training masks. Note that all masks should be resized with the same height before being averaged.

(ii) E-Step: Randomly sample a set of pixels $X = \{x_i\}$ for density estimation. In experiments, we sample 5% of the pixels from the window.

(iii) M-Step: Update the foreground and background probabilities for each pixel as follows:

$$F^t(y) = cF^{t-1}(y) \sum_{x_i \in X} F^{t-1}(x_i) \prod_{j=1}^{d} ker(y_j - x_{ij}) \tag{2}$$

$$B^t(y) = cB^{t-1}(y) \sum_{x_i \in X} B^{t-1}(x_i) \prod_{j=1}^{d} ker(y_j - x_{ij}) \tag{3}$$

Here c is a normalization coefficient, making sure $F^t(y) + B^t(y) = 1$.

(iv) Repeat E-Step and M-Step with several iterations.

In experiments, we reduce the windows by half before performing KDE-EM for speeding up, and finally enlarge the result maps twice. KDE-EM is performed in rgs color space[b] rather than the original RGB color space. This is because rgs color space is more robust to shade effect. The 3-dimensional feature vector is (r, g, s). The kernel function takes the Gaussian kernel: $ker(x) = exp(-0.5(x/\sigma)^2)$. The bandwidth is defined as: $\sigma_j = 1.06\hat{\sigma_j}|X|^{-0.2}$, $\hat{\sigma_j}$ is the standard deviation of the jth color channel. The iteration number N is set to three. Figures 2(c)–2(e) demonstrate the immediate results generated by KDE-EM.

As shown in Fig. 2(e), segmenting pedestrian by directly comparing F^N and B^N will result in noises. We use guided filtering[30] to remove noises. The main idea of guided filtering[30] is that the filter output q is locally linear to the guidance map I, i.e., $q_i = a_x I_i + b_x, \forall i \in w_x$, where w_x is the window with radius r centered at the pixel x. By minimizing the difference between the filter input p and output q, i.e., $Err(a_x, b_x) = \sum_{i \in w_x}((p_i - q_i)^2 - \varepsilon a_x^2)$, we can obtain a_x, b_x and q. Based on guided filtering,[30] we set the parameters $r = 4$, $\varepsilon = 0.01$ and perform local

[b] $r = R/(R + G + B)$, $g = G/(R + G + B)$ and $s = (R + G + B)/3$.

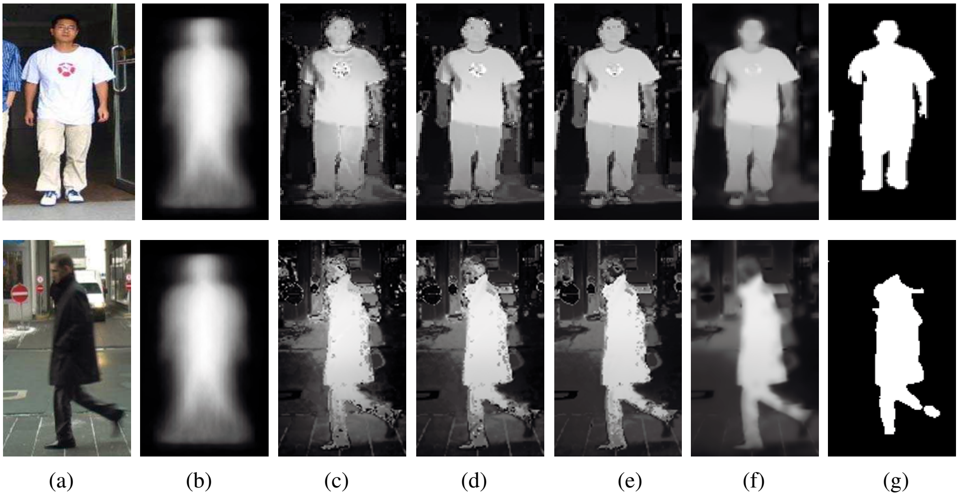


Fig. 2. (Color online) The initial pedestrian segmentation by KDE-EM. (a) The detection windows; (b) the shape priors; (c)–(e) the foreground probability maps produced at the 1st, 2nd and 3rd iteration; (f) the smoothed maps obtained by guided filtering; (g) the segmentation results.

smoothness with two steps: (1) taking the input image I as the guidance map, the foreground and background probability maps F^N, B^N are filtered into $\hat{F}^N$, $\hat{B}^N$ respectively; (2) the refined foreground and background probability maps are given by $F(y) = \hat{F}^N(y)/(\hat{F}^N(y) + \hat{B}^N(y))$, $B(y) = \hat{B}^N(y)/(\hat{F}^N(y) + \hat{B}^N(y))$. Figure 2(f) shows the smoothed result of Fig. 2(e). As can be seen, this strategy can significantly improve segmentation quality.

4. Shape Guided Puzzle Integration

While the results obtained by KDE-EM look decent in some cases, the human contours it generates are seldom aligned with image boundaries (see Fig. 2(g)). Furthermore, it fails in cases when the color models of the figure and ground are not highly discriminative. In order to solve these limitations, we introduce a shape guided puzzle integration scheme in this section. We first find the best matched template to the detected pedestrian from a set of shape templates, and then produce an assembly of puzzles that looks like the matched template. For speeding up, the shape templates are organized into a hierarchical tree. The puzzle generation algorithm employed in this paper is gPb-OWT-UCM,[31] which achieves state-of-the-art performance both on the general purpose segmentation and boundary detection benchmark.[c] The algorithm returns as output a hierarchical segmentation represented by a weighted edge map (see Fig. 3(b)). Using the puzzles at the lowest level as the computation units, we found that most of the boundaries in the original image are preserved.

[c] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

4.1. *Energy formulation for segmentation*

Let the puzzle set in the image be $S = \{s_i\}$ and the shape template be ST. The goal of pedestrian segmentation is to find the labeling configuration $L = \{l_i\}$ with maximum posterior probability(MAP): $L^* = argmax_L p(L|S, ST)$ where $l_i \in \{0, 1\}$, $l_i = 1$ represents that the puzzle s_i belongs to the foreground and vice versa.

Based on the Bayesian perspective, $p(L|S, ST) \propto p(S|L, ST)p(L)$, where $p(S|L, ST)$ is the observation likelihood of the image evidence, $p(L)$ is the labeling prior. We assume that the labeling prior follows the uniform distribution, and hence $L^* = argmax_L p(S|L, ST)$. This joint probability can be rewritten using an energy function $E(L)$, $p(S|L, ST) = -log(E(L))$. Since the puzzles satisfy Markovian property, the probability follows the Gibbs distribution in Markov random field(MRF).[32] We define the corresponding energy function as the summation of clique potentials:

$$E(L) = \lambda_1 \frac{1}{|S|} \sum_{s_i \in S} E_1(s_i|l_i) + \lambda_2 \frac{1}{|NB|} \sum_{(s_i, s_j) \in NB} E_2(s_i, s_j|l_i, l_j) \\ + \lambda_3 E_3(S|L, ST) \tag{4}$$

where NB represents pairs of adjacent puzzles, $(\lambda_1, \lambda_2, \lambda_3)$ are constant parameters weighting the proportion of the three potentials.

The unary potential $E_1(\cdot)$ is a low-level data term which imposes individual penalty for assigning any label l_i to the puzzle s_i. We rely on the appearance probability maps generated by KDE-EM to define the likelihoods:

$$E_1(s_i|l_i) = \begin{cases} \sum_{y \in s_i} F(y))/|s_i|\,, & l_i = 1 \\ \sum_{y \in s_i} B(y))/|s_i|\,, & l_i = 0 \end{cases} \tag{5}$$

The pairwise potential $E_2(\cdot)$, as the mid-level smooth term, defines to what extent adjacent puzzles should agree. It often depends on local observation. In our work, the pairwise potential has the form:

$$E_2(s_i, s_j|l_i, l_j) = \begin{cases} 1 - g(e_{s_i, s_j})\,, & l_i = l_j \\ g(e_{s_i, s_j})\,, & l_i \neq l_j \end{cases} \tag{6}$$

Here, e_{s_i, s_j} is the edgelet connecting the puzzle s_i and s_j, $g(e_{s_i, s_j})$ is the normalized gPb-OWT-UCM magnitude value of the edgelet e_{s_i, s_j}. This definition suggests that the adjacent puzzles should be assigned with different labels if their edgelet has a large magnitude value since the edgelet with a large magnitude value tends to lie in the figure-ground boundaries.

The shape guided potential $E_3(\cdot)$ is a high-level term, penalizing the labeling that is inconsistent with the shape template. We formulate $E_3(\cdot)$ as the summarization of the mask alignment terms $E_{3,m}(\cdot)$ and the boundary alignment terms $E_{3,b}(\cdot)$:

$$E_3(S|L, ST) = \frac{1}{|S|} \sum_{s_i \in S} E_{3,m}(s_i|l_i, ST) + \alpha_1 \frac{1}{|ST|} \sum_{t_i \in ST} E_{3,b}(S|L, t_i) \tag{7}$$

Let *Mask*(ST) be the mask generated by the shape template ST. The mask alignment term is the ratio for assigning s_i to the foreground or the background and defined by:

$$E_{3,m}(s_i|l_i, ST) = \begin{cases} \sqrt{\mathit{Area}(s_i \cap \mathit{Mask(ST)})/\mathit{Area}(s_i)}\,, & l_i = 1 \\ 1 - \sqrt{\mathit{Area}(s_i \cap \mathit{Mask(ST)})/\mathit{Area}(s_i)}\,, & l_i = 0 \end{cases} \tag{8}$$

Notably the template mask *Mask*(ST) should be resized with the same height to the labeling mask *Mask*(L) and placed at the center of *Mask*(L). Let *Edge*(L) be the boundaries produced by the labeling L. The boundary alignment term takes the Chamfer distance:

$$E_{3,b}(S|L, t_i) = min_{b_j \in Edge(L)}(min(|t_i - b_j|/\tau, 1) + \alpha_2 * g(b_j)) \tag{9}$$

Obviously, this term favors the labeling with strong boundaries and encourages that the labeling boundaries are aligned with the shape template. In the above formulas, τ is a truncating value, α_1, α_2 are weighting values.

4.2. *Energy inference by greedy puzzle merging*

To find the assembly minimizing the energy function $E(\cdot)$, i.e., $L^* = argmin_L E(L)$, one route is to search all assemblies. Yet it is computationally infeasible since we need to consider $2^{|S|}$ candidates, where $|S|$ is the puzzle number. Without constraint of the high-level potential $E_3(\cdot)$, the energy function is a well-known pairwise MRF-MAP problem and can be efficiently inferred by Graphcut[33] or Belief propagation.[34] Here, we take a greedy scheme to approximately optimize such a high-order energy function. Starting from the KDE-EM based initialization, we successively carry out a greedy puzzle merging operation by appending candidate puzzles to the foreground or background region. The merge operation works as a best-first search which only considers adding puzzles adjacent to the foreground and background regions and scores them via the energy change. The puzzles generated in the merging process are added to a candidate puzzle shortlist. The search terminates when the shortlist is empty. We summarize the greedy puzzle merging scheme as follows:

(i) **Initialization**

For the puzzle set $S = \{s_i\}$, the initial labeling $L^0 = \{l_i^0\}$ is given by:

$$l_i^0 = \begin{cases} 0\,, & F(s_i) > 0.5 \\ 1\,, & F(s_i) \le 0.5 \end{cases} \tag{10}$$

where $F(s_i) = \sum_{y \in s_i} F(y)/|s_i|$ is the mean KDE-EM based foreground probability for the puzzle s_i.

Based on the initial labeling L^0, we compute the initial energy $E^0 = E(L^0)$, obtain the foreground and background regions, and generate a candidate puzzle shortlist $Z = \{s_j\}$. Each candidate puzzle s_j should be adjacent to the foreground or background regions and satisfy *Score*$(s_j) > 0$. The score *Score*(s_j)

represents the energy change via altering the label of the puzzle s_j, i.e., $Score(s_j) = E^0 - E(\{\ldots, l^0_{j-1}, 1 - l^0_j, l^0_{j+1}, \ldots\})$.

(ii) **Puzzle merging**

Select the candidate puzzle with maximum score from the shortlist, $\hat{s} = argmax_{s_j} Score(s_j)$;

Update the labeling, $l^t_i = \begin{cases} l^{t-1}_i, & s_i \neq \hat{s} \\ 1 - l^{t-1}_i, & s_i = \hat{s} \end{cases}$ where t is the iteration number;

Merge the puzzle $\hat{s}$ to the foreground if its updated label is 1 or the background if its updated label is 0;

Recompute the labeling energy, $E^t = E(L^t)$.

(iii) **The shortlist updating**

Remove $\hat{s}$ from the shortlist;

For each puzzle s_k adjacent to the candidate $\hat{s}$, compute the score $Score(s_k)$:

$Score(s_k) = E^t - E(\{\ldots, l^t_{k-1}, 1 - l^t_k, l^t_{k+1}, \ldots\})$

Update the shortlist in terms of three cases:

(a) If $Score(s_k) > 0$ and s_k is within the shortlist, reset the score of s_k;

(b) If $Score(s_k) > 0$ and s_k is not within the shortlist, add s_k to the shortlist;

(c) If $Score(s_k) < 0$ and s_k is within the shortlist, remove s_k from the shortlist.

(iv) Repeat Steps (ii) and (iii) until the shortlist is empty.

Figure 3 demonstrates two examples of pedestrian segmentation via shape guided puzzle integration. As can be seen, the foreground pedestrians are gradually formulated as the energy is minimized.

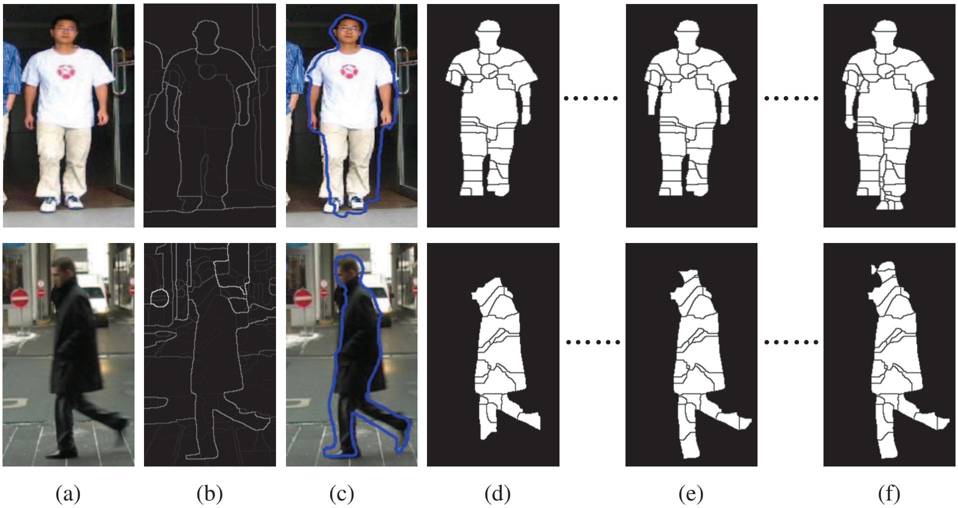


Fig. 3. (Color online) Examples of silhouette extraction by puzzle integration. (a) The input images; (b) the edge maps obtained by gPb-OWT-UCM; (c) the matched templates are overlaid on the images; (d)–(f) puzzles are gradually integrated into human-like silhouettes with the guide of the matched templates.

4.3. Construction of the hierarchical shape tree

As pedestrian shapes are highly variant, a set of shape templates is used to search the best matched template. For efficiency, those templates are organized in a hierarchical tree, in which similar templates are grouped together and represented with a prototype, as shown in Fig. 4. Shape matching is implemented as a process of traversing the tree to find the best matched prototype. Once the matching distance with a prototype is above a threshold, its following subtrees will not be visited, thus a significant speed-up can be achieved.

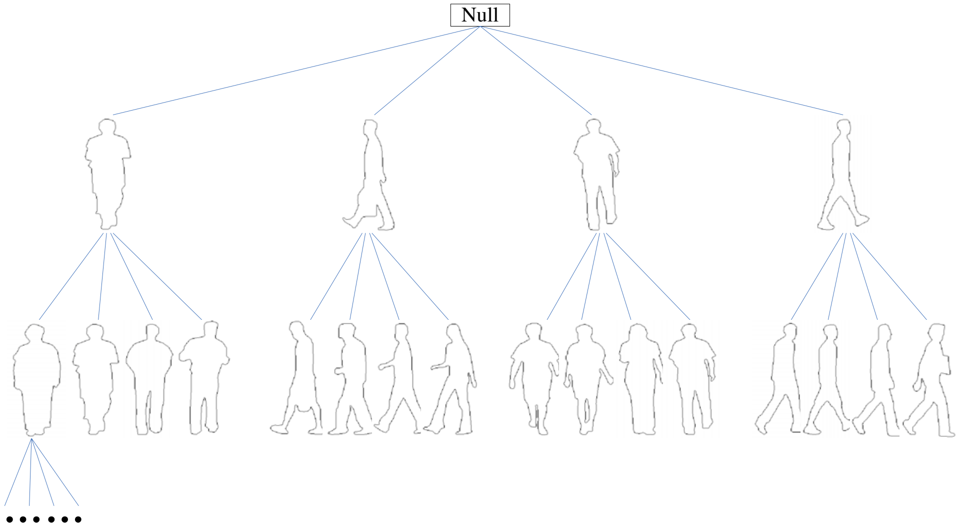


Fig. 4. The hierarchical shape tree.

Taking each shape template as a node of an Undirected Complete Graph (UCG) $G = \langle \hat{V}, \hat{E}, \hat{W} \rangle$, the construction of the tree can be considered as a problem of hierarchical graph clustering. This is a well-studied NP-hard problem in graph theory, involving some bottom-up clustering methods[1,35] and top-down partition methods.[36] Here, following the theory of spectral clustering,[36] we construct the hierarchical tree in a top-down manner. For the UCG $G = \langle \hat{V}, \hat{E}, \hat{W} \rangle$, we first calculate the edge weight matrix $\hat{W} = \{\hat{w}(i,j) | i,j \in \hat{V}, (i,j) \in \hat{E}\}$. The entity of the matrix $\hat{W}$ is defined as : $\hat{w}(i,j) = d(i,j) + d(j,i)$. $d(i,j)$ is the Chamfer distance between the template shape i and the template mask j (see Eq. (9)). $d(j,i)$ is obtained similarly. Spectral clustering partitions a graph into K subsets based on the normalize cut criterion:

$$Ncut_K = \sum_{i=1}^{K} \frac{cut(A_i, \hat{V} - A_i)}{assoc(A_i, \hat{V})} \tag{11}$$

where $assoc(A,B) = cut(A,B) = \sum_{u \in A, v \in B} \hat{w}(u,v)$.

Shi *et al.*[36] prove that an approximate solution can be obtained using the K eigenvectors of the K largest eigenvalues in: $\hat{W}\hat{X} = \hat{\lambda}\hat{D}\hat{W}$. Here $\hat{X}$ is the eigenvector matrix, $\hat{\lambda}$ is the eigenvalue matrix and $\hat{D}(i,i) = \sum_j \hat{w}(i,j)$ is a diagonal matrix. We utilize this approximate solution to iteratively divide the graph to construct a hierarchical tree. At first, nodes in the graph G are divided into K_1 subsets. Then, for each subset, the spectral clustering is employed again to partition it into K_2 sub-subsets. The process is recursively implemented until the number of clustering nodes is lower than a constant value K_3. The prototype of a subtree takes the template with the smallest mean similarity score to other templates in the subset. Taking each subset with its prototype as a subtree, the hierarchical tree is constructed.

Shape matching is applied as a coarse-to-fine traversal along the tree. At the non-leaf level, it is the prototypes derived to be aligned with the edge map, whereas, at the leaf level, all template exemplars are to be matched. If the matching distance of a prototype is above a threshold, all of its subtrees would not be visited, otherwise, the prototype is added to the list and the subtrees are traversed recursively. At last, we choose the template with the minimum distance in the visiting list as the best matched shape.

5. Constraint Pedestrian Refinement

Due to pose variances, pedestrian silhouettes produced in the puzzle integration stage are inaccurate in some cases. Further, puzzles may merge the foreground and background regions together. In this section, we refine pedestrian segmentation in pixel level with appearance cue. Comparing with the interactive figure-ground separation methods, e.g., Ref. 4, this refinement is automatically performed through the generated trimap which encodes pedestrian silhouette and skeleton information. In Section 5.1, we describe how to estimate the skeleton and the trimap. In Section 5.2, we state the pedestrian refinement procedure.

5.1. *Skeleton and trimap generation*

Pedestrian skeleton is composed of a set of line segments, each being connected by two joints, indicating the head, torso, upper/lower leg parts, as shown in Fig. 5(b).

In the learning phase, we manually click joints in the shape masks to yield the skeletons. For each point in the skeleton, we calculate its normal line and obtain the left and right crossing points between the normal line and the mask contour, resulting in a set $\{sp_i, lp_i, rp_i\}$. Here, sp_i is the skeleton point, lp_i and rp_i are the left and right crossing points. In the testing phase, as we have obtained pedestrian silhouette in which each point is matched to a sample point of the template contour, the skeleton can be easily transferred from the template to the image under the guide of silhouette. The skeleton in the image is denoted by: $\{sp'_i, lp'_i, rp'_i\}$, where $sp'_i = lp'_i + r_1\|rp'_i - lp'_i\|$, lp'_i is the matched point of lp_i, rp'_i is the matched point of rp_i, r_1 is a pre-computed value and defined as: $r_1 = \|rp_i - sp_i\|/\|rp_i - lp_i\|$.

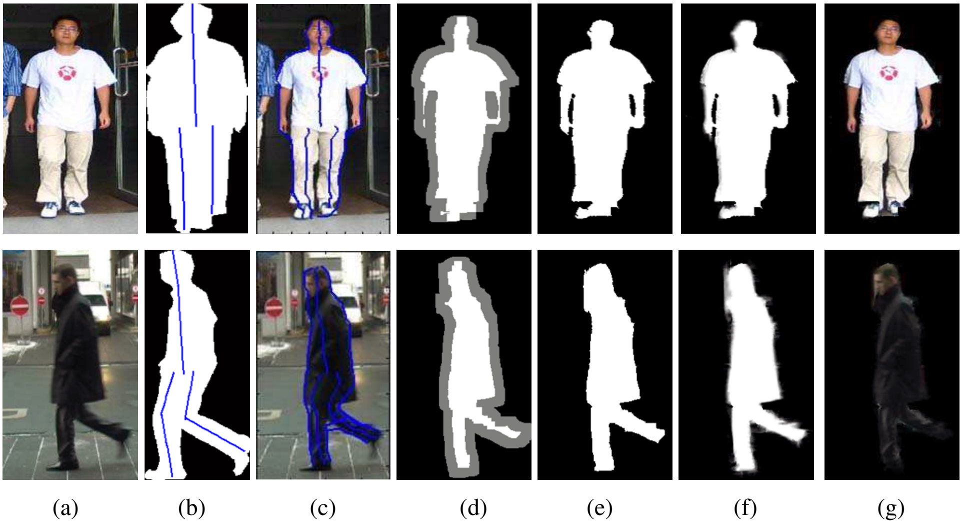


Fig. 5. (Color online) Examples of pedestrian refinement with the generated trimaps. (a) The input images; (b) the matched templates are overlaid with the skeletons; (c) the aligned silhouettes and the inferred skeletons on the original images; (d) the generated trimaps, in which the "Foreground" regions are denoted in white, the "Unknown" regions in gray and the "Background" regions in black; (e) the pedestrian masks obtained after refinement; (f) the alpha maps obtained after matting; (g) the final extracted pedestrians.

Based on the skeleton and silhouette point set $\{sp'_i, lp'_i, rp'_i\}$, the trimap in the detection window is automatically generated in the following way: For each pixel x, we first find a sample point in the skeleton with the minimum distance to it, i.e., $(D_{min}, k_{min}) = \min_i(|x - sp'_i|)$, where D_{min} is the minimum distance and k_{min} is the index. Then we compute the ratio $r_2 = \min(|x - lp'_{k_{min}}|, |x - rp'_{k_{min}}|)/D_{min}$. Given two thresholds T_1 and T_2 satisfying $0 < T_1 < T_2$, if $r_2 < T_1$, the pixel is assigned as "Foreground", and if $r_2 > T_2$, the pixel is assigned as "Background", otherwise, the pixel is assigned as "Unknown". So far a trimap is generated (see Fig. 5(d)).

5.2. *Human refinement via graphcut*

To refine the "Unknown" regions in the trimap, we follow the pairwise MAP-MRF formulation and define the corresponding energy function as follows:

$$E(\hat{L}) = \sum_{x_i \in U} E_d(\hat{l}(x_i)) + \lambda_4 \sum_{(x_i, x_j) \in EB} E_s(\hat{l}(x_i), \hat{l}(x_j)) \tag{12}$$

Here, E_d and E_s are the data and smooth terms respectively, U is the set of pixels in the "Unknown" regions, EB is the set of 4-neighboring pixel pairs, λ_4 is the weighting value. $\hat{L}$ is the labeling configuration, i.e., $\hat{L} = \{\hat{l}(x_i) | x_i \in U\}$, $\hat{l}(x_i) \in \{0, 1\}$ is the label assignment for the pixel x_i, $\hat{l}(x_i) = 0$ means the pixel x_i is assigned to the "Background", and $\hat{l}(x_i) = 1$ means to the "Foreground".

Several color models have been suggested for the definition of the data term, including K-Means, Histogram and Gaussian Mixture Model (GMM). We use GMM in our implementation. Based on the foreground and background regions of the trimap, two GMMs are estimated, one for the background and another for the foreground. Each GMM is taken to be a full-covariance Gaussian mixture with K_G components. The parameters of GMM are defined as: $\{(\mu_k^J, \Sigma_k^J)|k = 1, \dots, K_G, J \in \{B, F\}\}$, where (μ_k^F, Σ_k^F) are the mean and covariance for the foreground, and (μ_k^B, Σ_k^B) are for the background. For the pixels in the "Foreground" regions of the trimap, the data term is defined by: $E_d(\hat{l}(x_i) = 0) = 0$ and $E_d(\hat{l}(x_i) = 1) = \infty$. For the pixels in the "Background" regions of the trimap, the data term is defined by: $E_d(\hat{l}(x_i) = 0) = \infty$ and $E_d(\hat{l}(x_i) = 1) = 0$. For the pixels of the "Unknown" regions, the data term takes the form:

$$\begin{cases} E_d(\hat{l}(x_i) = 0) = d_i^F/(d_i^F + d_i^B) \\ E_d(\hat{l}(x_i) = 1) = d_i^B/(d_i^F + d_i^B) \end{cases} \tag{13}$$

where $d_i^J = \min_k \|(I(x_i) - \mu_k^J)' \ \Sigma_k^J (I(x_i) - \mu_k^J)\|$ is the similarity value between its color and the GMM components.

D_s is the smoothness term, which is given by:

$$E_s(l(x_i), l(x_j)) = \|I(x_i) - I(x_j)\|_2 |\hat{l}(x_i) - \hat{l}(x_j)| \tag{14}$$

This term encourages coherence in neighboring pixels with similar appearance.

An energy minimization solver — Graphcut[33] is applied to optimize $E(\hat{L})$ to obtain the refined pedestrian (as shown in Fig. 5(e)). As an initialization of Graphcut, the "Foreground" pixels are labeled as $\hat{l}(x_i) = 1$, the "Background" pixels are labeled as $\hat{l}(x_i) = 0$, and the "Unknown" pixels as $\hat{l}(x_i) = 0$ if $E_d(\hat{l}(x_i) = 0) > E_d(\hat{l}(x_i) = 1)$, or $\hat{l}(x_i) = 1$ if $E_d(\hat{l}(x_i) = 0) \leq E_d(\hat{l}(x_i) = 1)$. For further refining segmentation, we invoke the learning based matting method[37] to soft-segment an eroded narrow region along the boundaries (as shown in Figs. 5(f) and 5(g)).

6. Experimental Results

To illustrate the performance of the proposed approach, we apply it to several public datasets, including the Ethz sequences,[38] the Weizmann sequences,[39] the Fudan-Penn dataset,[14] the INRIA dataset[28] and the CALVIA dataset[d]. Both the Ethz[38] and the Weizmann[39] sequences capture pedestrians in their side walking. The Fudan-Penn dataset[14] contains humans in their nature pose. The INRIA[28] and CALVIN datasets focus on pedestrians in their front and rear views. The first three datasets have ground truth masks, which allow us to make quantitative comparison. We use the last two datasets to qualitatively demonstrate our method's robustness. In all datasets, the pedestrian windows are resized to 320 pixels in height, the width of the windows ranges from 85 pixels to 250 pixels.

[d] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1

Implementation Details. In the learning phase, we collect 423 shape templates from the Fudan-Penn pedestrian set.[14] All the templates are resized to 320 pixels in height and manually labeled on the joints. To construct the hierarchical shape tree, we set the parameters as $K_1 = 5$, $K_2 = 5$ and $K_3 = 10$, resulting in a 5-level tree. For a 210*320 pedestrian window, we typically extract about 250 puzzles. The other parameters set in experiments are kept the same, including: the truncating value $\tau = 20$, the weighting values $(\alpha_1, \alpha_2) = (0.5, 1.2)$, $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.3, 0.4, 0.3, 1.0/255.0)$, the threshold values $T_1 = 0.8$, $T_2 = 1.5$, $T_e = 1.6$, the Gaussian component number $K_G = 5$.

Segmentation Evaluation. For quantitatively evaluating our method's performance, we compare it with GrabCut,[4] SaliencyCut[18] and shape matching.[1] Both GrabCut[4] and SaliencyCut[18] segment pedestrians by optimizing MRF energy functions in pixel level using appearance cue. Shape matching[1] performs segmentation by matching the input image's edge map with a set of shape templates, thus only utilizes shape cue. SaliencyCut[18] and shape matching are automatic methods, while GrabCut[4] is an interactive method, requiring that users provide a bounding box for initialization and some scribbles for refinement. We perform GrabCut[4] on the original images and take the detection windows produced by PFF[7] as the bounding boxes to extract pedestrians.

The experiments are made on the Fudan,[14] Ethz[38] and Weizmann[39] datasets. The Fudan dataset[14] includes 160 pedestrians captured in real world, where pedestrians may be occluded by other objects or stand in cluttered backgrounds. The Ethz dataset[38] contains 5 pedestrian sequences, each sequence includes a walking circle with 12 images. The Weizmann pedestrians[39] are captured in controlled environments, having 18 walking sequences with 400 pedestrians in total. The segmentation accuracy is defined as the proportion of pixels correctly classified as foreground or background by comparing the binary segmentation result with the ground truth. We take the form: $\textit{F-measure} = 2 * \textit{precision} * \textit{recall}/(\textit{precision} + \textit{recall})$, where *precision* is defined as the ratio of the true positive pixels (i.e., the pixels labeled as foreground actually belong to foreground) to all labeled foreground pixels, and *recall* is defined as the ratio of the true positive pixels to ground truth pixels. Table 1 summarizes the results of performance evaluation over the three datasets.

Table 1. The mean segmentation accuracies (F-measure) obtained with GrabCut, SaliencyCut and shape matching over the Weizmann, Ethz and Fudan-Penn datasets.

Method	Weizmann[39]	Ethz[38]	Fudan[14]
GrabCut[4]	71.42%	66.11%	67.43%
SaliencyCut[18]	77.49%	65.02%	71.45%
Shape matching[1]	76.87%	78.07%	79.68%
Our method(Stage I)	81.15%	79.24%	78.18%
Our method(Stage II)	84.02%	84.22%	84.74%
Our method(Stage III)	88.04%	85.78%	85.30%

As can be seen, our method achieves better performance, improving about 18%, 15% and 8% over GrabCut,[4] SaliencyCut[18] and shape matching[1] respectively. The main contribution of our method is the combination of shape, puzzle and appearance cues for pedestrian segmentation. The benefit of such combination is also demonstrated in Table 1, which verifies that the accuracies are improved by combining the three cues together.

Figure 6 qualitatively compares the segmentation results of GrabCut,[4] SaliencyCut,[18] shape matching[1] and our method. We can observe that GrabCut[4] without scribbles and SaliencyCut[18] fail to produce human-like segmentation. This is expected as they only rely on local appearance and do not incorporate any high-level cues. Although GrabCut[4] with scribbles can produce more accurate segmentation, it requires cumbersome interactions and the results are sensitive to the interactions. Shape matching[1] can produce approximate pedestrian contours, but the contours are not aligned with body boundaries. For example, in the last row of Fig. 6, the two arms are missed. In contract, our method which takes advantage of the relative merits of high-level shape, middle-level puzzle and low-level appearance, is able to extract more accurate masks. As shown in Fig. 6(f), the initial stage — KDE-EM can approximately estimate figure-ground separations. Although such separations may not follow pedestrian silhouettes and produce erroneous regions, the second stage utilizing shape guided puzzle integration scheme can refine them, in which the puzzles are able to preserve local contour information and the shape template is able to complete the incomplete regions. The last stage — pedestrian refinement is very helpful in some faint figure-ground regions (see the last row of Fig. 6).

In order to further demonstrate the performance of our method, we compare our segmentation results with those obtained by Lin *et al.*[10] and Wu *et al.*[11] Both of them extract pedestrian silhouettes using local contours. Although local contours are more flexible than global shape templates, their methods still have three limitations: (1) constraint to frontal/rear view pedestrians; (2) ignoring the segmentation of arms; (3) the ambiguous delineation of local contours. The second row of Fig. 7 shows some examples of the segmentation results derived from their algorithms. In the third row of Fig. 7, we demonstrate the inferred segmentation of our approach. As we can see, our pedestrian extraction gives more accurate delineation of pedestrian silhouettes. The pedestrian arms are also segmented. In addition, our algorithm can be applied to segment pedestrians in side profile view.

Many pedestrians are partially or severely occluded in real scene. The segmentation task becomes difficult in those cases. In order to give a better impression of our method's robustness to occlusion, Fig. 8 therefore demonstrates the produced segmentation results on examples with different occlusion. As can be seen from those examples, our method can still segment pedestrians with partial occlusion, but fails with severe occlusion.

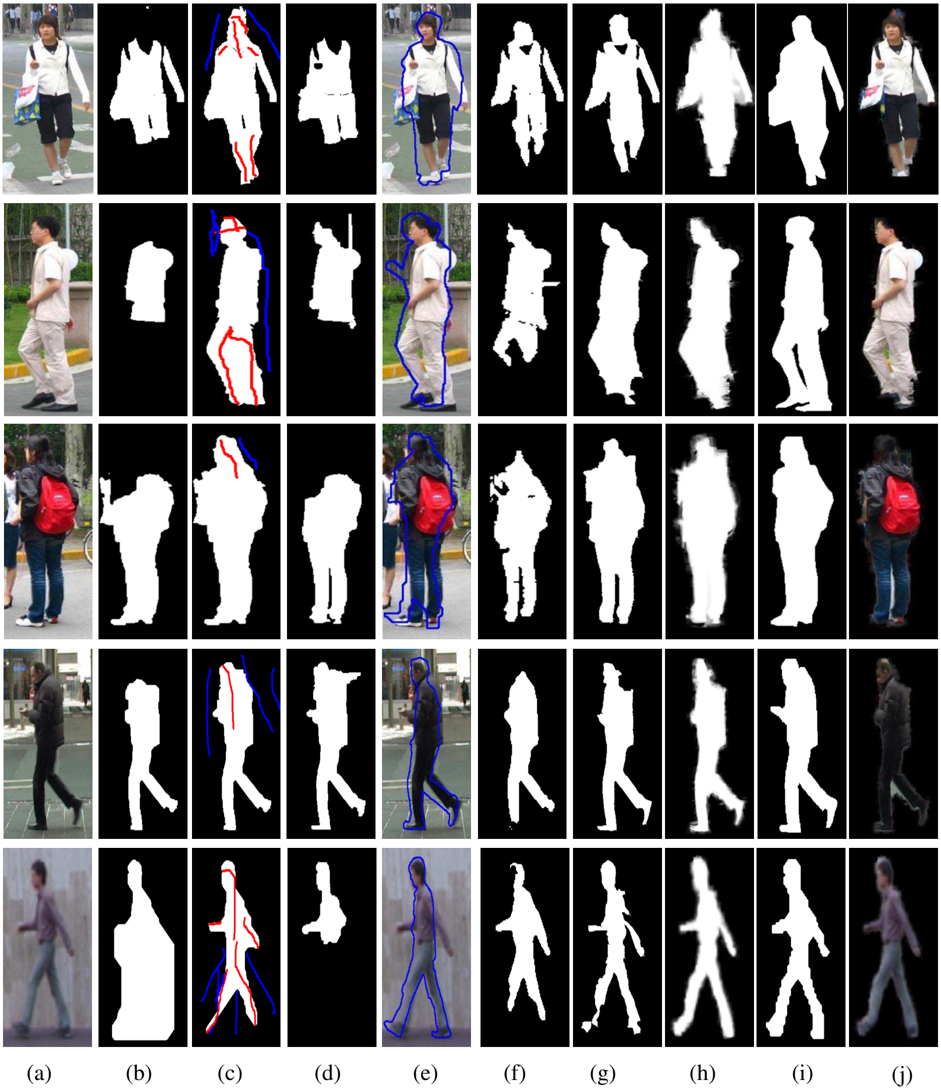
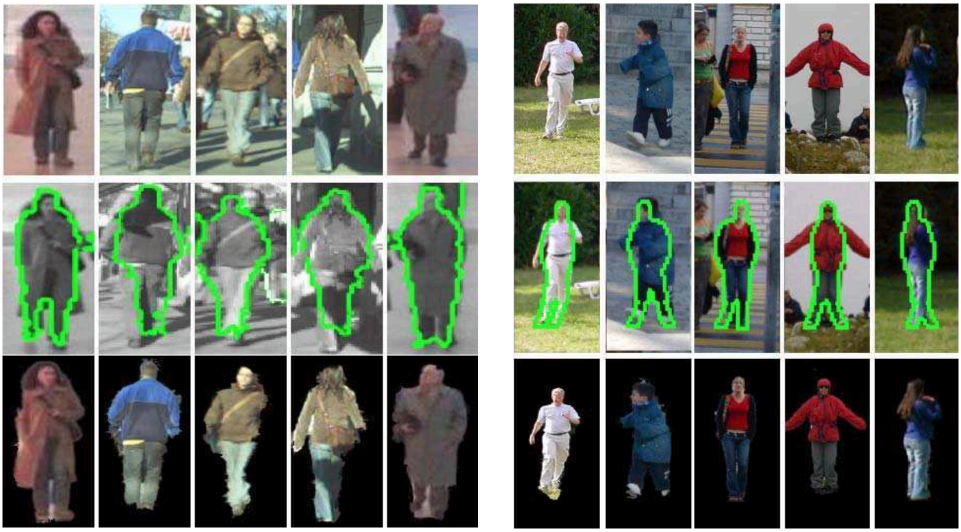


Fig. 6. (Color online) Comparison of segmentation results with GrabCut, SaliencyCut and shape matching. (a) The input images; (b) the segmentation results of GrabCut without scribbles; (c) the segmentation results of GrabCut with scribbles, in which the foreground (in red) and background (in blue) scribbles are overlaid on the segmentation results; (d) SaliencyCut's results; (e) shape matching's results, in which the silhouettes (in blue) produced by shape matching are overlaid on the input images; (f)(g)(h) the segmentation results produced by our method at stage I, stage II and stage III; (i) the groundtruth masks; (j) the final extracted pedestrians by our method.

Time Cost. Our experiments were implemented in Matlab with some employed functions, including PFF,[7] hierarchical segmentation,[31] guided filtering[30] and learning based matting.[37] The experiments were done on a computer with 2.3 GHz CPU and 3.0 GB RAM. In the learning phase, the construction of the hierarchical tree



(a)

(b)

Fig. 7. (Color online) Comparison of segmentation results with Wu *et al.* and Lin *et al.* (a) The segmentation results of our method and Wu *et al.* over the CAVIAR dataset; (b) The segmentation results of our method and Lin *et al.* over the INRIA dataset. The first row shows the input images from the CAVIAR and INRIA datasets. The second row displays the segmentation results of Wu *et al.* and Lin *et al.*, in which the extract silhouettes are displayed in green. The third row shows our results.

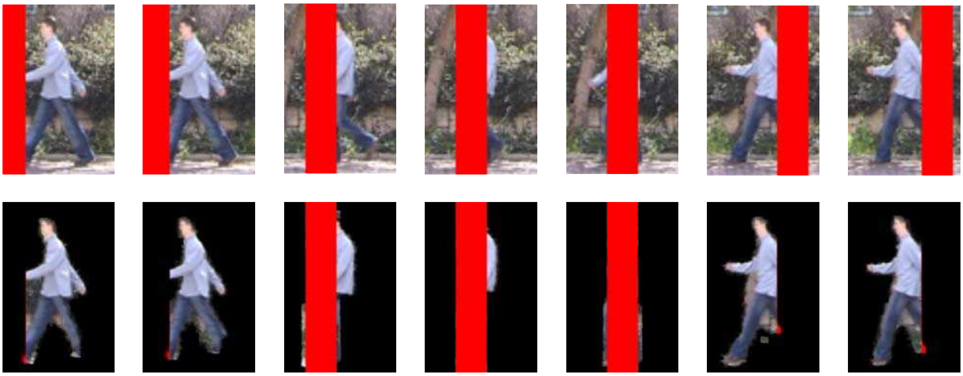


Fig. 8. (Color online) Examples of pedestrian segmentation under occlusion. The first row shows some input images, the second row shows the corresponding segmentation results.

takes about 6 hours, of which most of the time is spent on computing the weighting matrix $\hat{W}$.

To extract the pedestrian in a 210∗320 window, the initial stage takes about 6.8 s, of which KDE-EM takes 6.7 s and guided filtering takes 0.1 s. At stage II, puzzle generation costs 80 s. Considering that this step has a parallel version,[31] significant performance may be achieved. Shape matching with a template is performed in 0.3s. Since the hierarchical tree reduces the number of the matching templates from 423 to 25, the time is correspondingly reduced from 126.9 s to 7.5 s. The time of greedy puzzle integration is only 0.1 s. The stage III — pedestrian refinement takes 2.8 s. Thus, the total time is about 97.2 s per image, yet most of the time is spent on extracting puzzles, while the main processing in the paper, including KDE-EM, shape matching, greedy puzzle merging and pedestrian refinement cost approximately 17.2 s in total.

7. Conclusion

In this paper we present a solution for pedestrian segmentation in still images. The solution is cast in a three-stage framework using high-level shape, middle-level puzzle and low-level appearance cues. The first stage utilizes KDE-EM to extract pedestrian probabilities for initialization. The second stage performs pedestrian extraction via integrating puzzles with constraint of shape cue. The third stage refines segmentation within an appearance trimap which encodes both human silhouette and skeleton. Qualitative and quantitative results show that the combination of the three cues significantly improve pedestrian segmentation.

Although our approach can handle the majority of standing pedestrian segmentation, some mislabeled pixels still exist due to faint figure-ground differences or occlusion. Future work will consider improving it more robust to cluttered scenes. We are also investigating to extend it for video pedestrian segmentation by incorporating motion cue into this framework.

Acknowledgments

This work is supported by the National 863 Program of China under Grant No. 2012AA011803, the Natural Science Foundation of China under Grant No. 61170188 and the Innovation Foundation of BUAA for PhD Graduates under Grant No. 292136.

References

1. D. M. Gavrila, A bayesian exemplar-based approach to hierarchical shape matching, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **29**(2) (2007) 1408–1421, doi:10.1109/TPAMI.2007.1062.
2. B. Fulkerson, A. Vedaldi and S. Soatto, Class segmentation and object localization with superpixel neighborhoods, in *IEEE Int. Conf. on Computer Vision* (2009), pp. 670–677, doi:10.1109/ICCV.2009.5459175.
3. Y. J. Lee and K. Grauman, Collect-cut: Segmentation with top-down cues discovered in multi-object images, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2010), pp. 3185–3192, doi:10.1109/CVPR.2010.5539772.
4. C. Rother, V. Kolmogorov and A. Blake, "GrabCut": Interactive foreground extraction using iterated graph cuts, *ACM Transactions on Graphics* **23**(3) (2004) 309–314, doi:10.1145/1015706.1015720.

5. S. Maji, N. K. Vishnoi and J. Malik, Biased normalized cuts, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2011), pp. 2057–2064, doi:10.1109/CVPR.2011.5995630.
6. T. H. Kim, K. M. Lee and S. U. Lee, Learning full pairwise affinities for spectral segmentation, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2010), pp. 2101–2108, doi:10.1109/CVPR.2010.5539888.
7. P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan, Object detection with discriminatively trained part based models, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **32**(9) (2010) 1627–1645, doi:10.1109/TPAMI.2009.167.
8. L. Alvarez, L. Baumela, P. M. Neila and P. Henriquez, Morphological snakes, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2010), pp. 2197–2202, doi:10.1109/CVPR.2010.5539900.
9. A. Opelt, A. Pinz and A. Zisserman, A boundary-fragment-model for object detection, in *Proc. of European Conf. on Computer Vision* (2006), pp. 575–588, doi:10.1007/11744047_44.
10. Z. Lin and L. S. Davis, A pose-invariant descriptor for human detection and segmentation, in *Proc. of European Conf. on Computer Vision* (2008), pp. 423–436, doi:10.1007/978-3-540-88693-8_31.
11. B. Wu and R. Nevatia, Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses, *Int. J. of Computer Vision* **82**(2) (2009) 185–204, doi:10.1007/s11263-008-0194-9.
12. W. Gao, H. Ai and S. Lao, Adaptive contour features in oriented granular space for human detection and segmentation, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2009), pp. 1786–1793, doi:10.1109/CVPR.2009.5206762.
13. B. Leibe, E. Seemann and B. Schiele, Pedestrian detection in crowded scenes, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2005), pp. 878–885, doi: 10.1109/CVPR.2005.272.
14. L. Wang, J. Shi, G. Song and I. Shen, Object detection combining recognition and segmentation, in *Proc. of Asian Conf. on Computer Vision* (2007), pp. 189–199, doi:10.1007/978-3-540-76386-4_17.
15. D. Larlus and F. Jurie, Combining appearance models and markov random fields for category level object segmentation, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2008), pp. 1–8, doi:10.1109/CVPR.2008.4587453.
16. Z. Tu, Auto-context and its application to high-level vision tasks, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2008), pp. 1–8, doi:10.1109/CVPR.2008.4587436.
17. J. Wang and M. Cohen, Image and video matting: A survey, *Foundations and Trends in Computer Graphics and Vision* **3**(2) (2007) 97–175, doi:10.1561/0600000019.
18. M. Cheng, G. Zhang, N. J. Mitra, X. Huang and S. Hu, Global contrast based salient region detection, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2011), pp. 409–416, doi: 10.1109/CVPR.2011.5995344.
19. P. Mehrani and O. Veksler, Saliency segmentation based on learning and graph Cut refinement, *Proc. of British Machine Vision Conf.* (2010), pp. 1–12, doi:10.5244/C.24.110.
20. B. Alexe, T. Deselaers and V. Ferrari, ClassCut for unsupervised class segmentation, in *Proc. of European Conf. on Computer Vision* (2010), pp. 380–393, doi:10.1007/978-3-642-15555-0_28.
21. G. Mori, X. Ren, A. A. Efros and J. Malik, Recovering human body configurations: Combining segmentation and recognition, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2004), pp. 326–333, doi:10.1109/CVPR.2004.1315182.

22. Y. Bo and C. C. Fowlkes, Shape-based pedestrian parsing, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2011), pp. 2265-2272, doi:10.1109/CVPR.2011.5995609.
23. E. Borenstein, E. Sharon and S. Ullman, Combining top-down and bottom-up segmentation, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **30**(12) (2008) 2109–2125, doi:10.1109/TPAMI.2007.70840.
24. A. Levin and Y. Weiss, Learning to combine bottom-up and top-down segmentation, in *Proc. of European Conf. on Computer Vision* (2006), pp. 581–594, doi:10.1007/11744085_45.
25. M. P. Kumar, P. Torr and A. Zisserman, OBJ CUT, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2005), pp. 18–25, doi:10.1109/CVPR.2005.249.
26. M. Bray, P. Kohli and P. Torr, PoseCut: Simultaneous segmentation and 3D pose estimation of humans using dynamic graph-Cuts, in *Proc. of European Conf. on Computer Vision* (2006), pp. 642–655, doi:10.1007/11744047_49.
27. Z. Lin, L. S. Davis, D. Doermann and D. DeMenthon, An interactive approach to pose-assisted and appearance-based segmentation of humans, in *IEEE Int. Conf. on Computer Vision* (2007), pp. 1–8, doi:10.1109/ICCV.2007.4409123.
28. N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2005), pp. 886–893, doi:10.1109/CVPR.2005.177.
29. D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization* (Wiley Interscience, New York, 1992).
30. K. He, J. Sun and X. Tang, Guided image filtering, in *Proc. of European Conf. on Computer Vision* (2010), pp. 1–14, doi:10.1007/978-3-642-15549-9_1.
31. P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, From contours to regions: An empirical evaluation, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2009), pp. 2294–2301, doi:10.1109/CVPRW.2009.5206707.
32. S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 3rd edn. (Springer-Verlag, New York, 2009).
33. Y. Boykov, O. Veksler and R. Zabih, Fast approximate energy minimization via graph cuts, in *IEEE Int. Conf. on Computer Vision* (1999), pp. 377–384, doi:10.1109/ICCV.1999.791245.
34. P. F. Felzenszwalb and D. P. Huttenlocher, Efficient belief propagation for early vision, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2004), pp. 261–268, doi:10.1109/CVPR.2004.1315041.
35. P. F. Felzenszwalb and D. P. Huttenlocher, Efficient graph-based image segmentation, *Int. J. of Computer Vision* **59**(2) (2004) 167–181, doi:10.1023/B:VISI.0000022288.19776.77.
36. J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **22**(8) (2000) 731-737, doi:10.1109/34.868688.
37. Y. Zheng and C. Kambhamettu, Learning based digital matting, in *IEEE Int. Conf. on Computer Vision* (2009), pp. 889–896, doi: 10.1109/ICCV.2009.5459326.
38. M. Andriluka, S. Roth and B. Schiele, People-tracking-by-detection and people-detection-by-tracking, in *IEEE Int. Conf. on Computer Vision and Pattern Recognition* (2008), pp. 1–8, doi:10.1109/CVPR.2008.4587453.
39. L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri, Actions as space-time shapes, in *IEEE Int. Conf. on Computer Vision* (2005), pp. 1395–1402, doi:10.1109/ICCV.2005.28.